



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/717,537	11/21/2000	Christopher G. Kaler	777.335US1	1797

7590

03/23/2004

Steven J Rocci  
Woodcock Washburn Kurtz Mackiewicz & Norris LLP  
One Liberty Place  
46th Floor  
Philadelphia, PA 19103

EXAMINER

VU, TUAN A

ART UNIT PAPER NUMBER

2124

DATE MAILED: 03/23/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

# Office Action Summary

Application No.

09/717,537

Applicant(s)

KALER ET AL.

Examiner

Tuan A Vu

Art Unit

2124

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

## Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

## Status

- 1) ☒ Responsive to communication(s) filed on 15 January 2004.
- 2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

## Disposition of Claims

- 4) ☒ Claim(s) 1-19 and 21-66 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☒ Claim(s) 63-66 is/are allowed.
- 6) ☒ Claim(s) 1-19 and 21-31, 33-62 is/are rejected.
- 7) ☒ Claim(s) 32 is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

## Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on \_\_\_\_\_ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

## Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some \* c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
2. ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).
- \* See the attached detailed Office action for a list of the certified copies not received.

## Attachment(s)

- 1) ☐ Notice of References Cited (PTO-892)
- 2) ☒ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)  
Paper No(s)/Mail Date \_\_\_\_\_.
- 4) ☐ Interview Summary (PTO-413)  
Paper No(s)/Mail Date \_\_\_\_\_.
- 5) ☐ Notice of Informal Patent Application (PTO-152)
- 6) ☐ Other: \_\_\_\_\_.

**; DETAILED ACTION**

1. This action is responsive to the Applicant's response filed 1/15/2004.

As indicated in Applicant's response, claims 12, 25 have been amended, claim 20 canceled. Claims 1-19, 21-66 are pending in the office action.

***Claim Rejections - 35 USC § 102***

2. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

Note: 35 U.S.C. § 102(e), as revised by the AIPA and H.R. 2215, applies to all qualifying references, except when the reference is a U.S. patent resulting directly or indirectly from an international application filed before November 29, 2000. For such patents, the prior art date is determined under 35 U.S.C. § 102(e) as it existed prior to the amendment by the AIPA (pre-AIPA 35 U.S.C. § 102(e)).

3. Claims 1-10, 12-31, 33-38, 41-60, and 62 are rejected under 35 U.S.C. 102(e) as being anticipated by Underwood, USPN: 6,523,027 ( hereinafter Underwood).

**As per claim 1**, Underwood discloses a system for developing software, comprising:

a version store for storing a plurality of development documents ( e.g. Fig. 93A-B; col. 252, lines 21-33);

a plurality of service providers performing different development functions, and each having a mutually compatible interface( e.g. *user interface, activities, ASP, views* – Fig. 47; presentation- Fig. 48; *RETASRV1, RETADB1, RETASRV2, RETADEV1* - Fig. 40; *Web & Application Server, Netscape communicator, Internet Explorer* - Figs. 51-53; *console Site server*

Art Unit: 2124

– Fig. 60-64; Fig. 56-67; Fig. 72B – Note: interface could be any flavor of windows-based screens or browser technologies to facilitate passage of commands or display of content and more than one servers amounts to a set of service providers communicating under the protocol of web based technologies, for example IE, and the mutual compatibility of such interfaces are inherent otherwise no communication could be effected);

a client having the same compatible interface (e.g. col. 30, lines 35-63; Fig. 103B; developer workstation – Fig. 40, 65-67), for receiving commands from a user and for routing them to various ones of the service providers for execution upon development documents from the version store ( e.g. *first server, second server* – Fig. 1C; Fig. 30-31, 36, 39-42; *router, Wan* – Fig. 124-129; Fig. 106-118– Note: the limitation of routing is implicitly disclosed in view of the protocol for windows-based web applications in a Lan or Wan or multi-servers networking context; and use of screen command to effect sourceSafe tool discloses passing of commands to be routed to destined servers storing versioned data).

**As per claim 2**, Underwood discloses a third party (e.g. col. 293, lines 7-15; *backup router* - Fig. 127; Fig. 128; *Alta Vista* - col. 315, lines 20-34; col. 289, lines 20-67; col. 91, line 55 to col. 106, line 28 – Note: all products recommended for Reta tool not made by Reta are third party products/services, e.g. Visual Sourcesafe, Oracle client, PVCS, MOM, Rationale Rose, Visual Studio).

**As per claim 3**, see Underwood : Fig. 26, Fig. 93B, Fig. 73, 94( Note: source repository in conjunction with ODBC server is equivalent to version database/store)

**As per claim 4**, Underwood discloses API and set of methods (e.g. col. 129, lines 15-27; col. 130, line 55 to col. 132, line 14 -- Note: use of browser and script and windows GUI

Art Unit: 2124

embedded functions are equivalent to using API; *JavaBean*, *ActiveX* - col. 332, lines 37-48; *AFVBAActivityWrapper* - col. 35, line 56 to col. 37, line 21; col. 37, line 24 to col. 38, line 35).

**As per claim 5**, see Underwood: col. 121, line 40 to col. 126, line 44 ( Note: the diversity of services to implement ReTa implicitly discloses that each service providers implements less than all the method in a set of methods called for in the browser session).

**As per claim 6**, see Underwood col. 129, lines 15-27; *OLE DB* - col. 153, *Database Connectivity*, Description b/w step 8 and step 9).

**As per claim 7**, Underwood discloses object model exposing the same compatible interface ( e.g. Fig. 31, 104-105, Fig. 110, 109A-B, 113-114; Fig. 117 – Note: source save view with model and tree structure to lay out the customer queries and templates is equivalent to object model exposing the interface; Fig. 36, 122 – Note: Rational Rose modeling tool to lay out the user interfacing activity with the required object is equivalent to model exposing the interface).

**As per claim 8**, Underwood discloses a command-line utility (e.g. Fig. 68-71; col. 187, line 67 ).

**As per claim 9**, see Underwood: Fig. 72B, 73-74.

**As per claim 10**, Underwood discloses merge subsystem ( e.g. *merge* - col. 90, line 59 to col. 91, line 14)

**As per claim 12**, Underwood discloses a service provider for a software development system, comprising:

Art Unit: 2124

code for performing a development service in a software development system (e.g. Fig. 65-67, 68-71; Fig. 72A; Fig. 74-75; Fig. 78-83B – Note: service to debug and support change tracking are equivalent to code for software development service);

an interface for communicating with a set of further service providers (e.g. *Workstations and Windows, Netscape, I explorer, Oracle client/enterprise, database server and web server --* Fig. 65-66 – Note: interface could be any flavor of windows-based screens or browser technologies to facilitate passage of commands or display of content and more than one servers amounts to a set of service providers ) and with a client for receiving development commands from a user (e.g. Fig. 10-14; Fig. 72B; Fig. 77-82 – Note: use of windows-based screens or browser panes implicitly disclose text fields wherein commands or mouse clicks can be processed).

**As per claims 13 and 14**, Underwood discloses a version store for storing and retrieving development documents and processing one or more thereof (e.g. Fig. 93A-B; SourceSafe - Fig. 110 – Note: check in files into version control repository is equivalent to version store).

**As per claim 15**, Underwood discloses query processing service (e.g. col. 123, line 56 to col. 124, line 21; col. 129, lines 5-57).

**As per claim 16**, Underwood discloses enlistment management (e.g. Activity Framework: *BOMapping, AFCollection, AFUIList, EventHandler, PersistableObj, SystemPreferences, TrackingManager --* cols. 35 – 62; Fig. 18-21).

**As per claim 17**, Underwood discloses interface for holding, storing and retrieving copies of documents ( e.g. col. 37, lines 30-33; Fig. 64, 111; col. 164, lines 16-18; Fig. 93A).

**As per claim 18**, Underwood discloses compatible interface ( e.g. *Activity* – Fig. 15B; *SubActivity* – Fig. 17B; Fig. 19-22 – Note: server communicating with activity context of client implicitly discloses interface compatibility) for communicating with further providers (e.g. *first server, second server* – Fig. 1C; Fig. 30-31, 36, 39-42).

**As per claim 19**, see Fig. 93A.

**As per claim 21**, refer to claim 4 for corresponding rejection, respectively.

**As per claim 22**, see claim 5.

**As per claim 23**, Underwood discloses providers providing subsets of methods ( e.g. Fig. 65; Fig. 72B; Fig. 93-101 – Note: each module performed by each team in the configuration process is equivalent to subset of methods; col. 119, line 35 to col. 122, line 67 - Note: each service performing a subset of the Framework runtime services is equivalent to subset of methods).

**As per claim 24**, refer to claim 6.

**As per claim 25**, Underwood discloses a client for a software development system having a plurality of service providers, said service providers development services comprising:

means for receiving commands from a user for executing development operations (e.g. Fig. 10-14; Fig. 72B; Fig. 77-82);

an interface for communicating with compatible interfaces of all the plurality of service providers (e.g. *user interface, activities, ASP, views* – Fig. 47; *presentation*- Fig. 48; *Web & Application Server, Netscape communicator, Internet Explorer* - Figs. 51-53; *console Site server* – Fig. 60-64; Fig. 56-67; Fig. 72B – Note: more than one server amounts to a set of service providers providing services and further disclose compatible interfaces which are the likes of

Art Unit: 2124

windows-based interfaces under web-based protocol, the compatibility of such communication interface being inherently compatible with the mutually communicating services).

**As per claim 26**, Underwood discloses means for accessing and processing a plurality of enumerators (e.g. col. 29, lines 5-15; col. 35, lines 30-54; Figs. 103A, 147 – Note: Collection and marshalling as well as analysis/browsing of HTML/ASP/Scripts are equivalent to enumerator; Fig. 101-118 – Note: SourceSafe display of items is equivalent to enumerating and mapping to user's criteria).

**As per claim 27**, Underwood discloses a merge subsystem (see claim 10; col. 258, lines 50-66).

**As per claim 28**, Underwood discloses a plurality of preprocessors for receiving and modifying the input stream representing a plurality of input development documents (e.g. Fig. 94—103A; Fig. 128, 145B, 149, 151 – Note: security server and version control servers are equivalent to preprocessors); a merge engine for outputting a merge document (re claim 10; col. 258, lines 50-66 – Note: preprocessors to route the request for merge to PVCS server are equivalent to preprocessors for merge subsystem).

**As per claim 29**, Underwood discloses a selection of hosts and servers and security services (e.g. col. 318, lines 47-67; col. 289, lines 28-60; *Routers* – col. 311, line 46 to col. 312, line 12; Fig. 124-127; *Workbench*, *SourceSafe Administrator*, *SourceSafe Explorer*, *Queries*, *Tracking*, *Reports* – Fig. 104-117 – Note: each processes called upon by the GUI events to retrieve, modify and reconcile documents for a development process is equivalent to a selected processing element preceding or following another processing element); hence discloses a



Art Unit: 2124

selection service for processors or preprocessors, i.e. a brokers, some of which, implicitly, can be merge subsystem processors.

**As per claim 30**, a plurality of postprocessors outputting a merge document is implicitly implied from the merge processing service as disclosed in claims 28 and 10 ( Note: a server operable (PVCS server) to yield a merged document is a postprocessor while the preprocessors are those who take the input for the merging operation; and the plurality of processors to take the output back to the client machine are the same processors in the path directing the input going to the merging machine).

**As per claim 31**, Underwood discloses keyword processing (*ByRef* - col. 298, lines 24-45)

**As per claim 33**, refer to corresponding rejection set forth in claim 13.

**As per claim 34**, Underwood discloses a merge system for a software development system, comprising: a merge engine for merging development documents into a merge document (e.g. *merge* - col. 90, line 59 to col. 91, line 14; *NoActionsMerge* - col. 258, lines 50-66); a merge command (e.g. col. 90, line 59 to col. 91; Fig. 106-118 – Note: merge command is implicit to PVCS commands); a plurality of merge preprocessors (see claim 28); and a merge broker (see claim 29).

**As per claim 35**, see Underwood: Fig. 104-118.

**As per claim 36**, this claim is inherent to selection of processors in claim 29 above.

**As per claim 37-38**, these claims limitations are implicitly disclosed via claim 29 and 30.

**As per claim 41**, Underwood discloses a method for developing software using versioned documents in a programmed digital computer, comprising:

Art Unit: 2124

sending commands from a client through a versioning interface (e.g. Fig. 93a; Fig. 103A-118);

receiving the commands in a plurality of service providers all having an interface compatible with the versioning interface (e.g. Fig. 75-85; Fig. 93A-103A; Fig. 104-118);

executing the commands in the providers (e.g. Fig. 86, 98-103A, 111-116; Fig. 132A – Note: SourceSafe service checking in of documents is equivalent to processing of user commands);

retrieving and storing a plurality of development documents in a version store (Note: this is inherent to SourceSafe service from above), while executing the commands.

**As per claim 42**, this claim is implicitly disclosed by virtue of corresponding rejections addressing limitations in claim 1 such as plurality of providers and routing, and claims 29 or 30.

**As per claim 43**, Underwood discloses an interface for retrieving different from interface for version store (e.g. *source repository and ODBC/assembly/test* – Fig. 42; Fig. 65).

**As per claim 44-46**, refer to corresponding rejections as set forth in claims 4-6, respectively.

**As per claims 47, 48, and 49**, Underwood discloses selecting providers and replacing providers; adding another provider (col. 289, lines 20-59; col. 311, line 46 to col. 312, line 12; Fig. 124-127 – Note: selecting providers or routing to alternate servers is equivalent to replacing and adding providers, the selecting of services to effect versioning and processing of documents is implied in this selecting and routing through gateways or firewalls); and replacing client with a third-party provider (e.g. col. 293, lines 7-15; *backup router* - Fig. 127; Fig. 128; *Alta Vista* - col. 315, lines 20-34 – Note: replacing standard browser security check with third-party security

Art Unit: 2124

service is equivalent to replacing client processes with commercial processes) with the same versioning interface.

**As per claim 50**, this is a medium version of claim 41, hence incorporates the same rejection as set forth in claim 41 above.

**As per claim 51**, Underwood discloses a method for developing software in a programmed digital computer, comprising:

receiving a user command from a client having a versioning interface in one of a plurality of service providers each having an interface compatible with the versioning interface (e.g. Fig. 75-85; Fig. 93A-103A; Fig. 104-118);

executing an operation in response to the user command (e.g. Fig. 86, 98-103A, 111-116; Fig. 132A; Fig. 134-144);

communicating a result through the interface (e.g. Fig. 76A-86; Fig. 134-144);

repeating the above steps for others of the services providers (e.g. Fig. 126-129; col. 293, lines 7-15; *backup router* - Fig. 127; Fig. 128; *Alta Vista* - col. 315, lines 20-34 -- Note: this step is implicitly disclosed via a selecting of commercial services, routing and web session/activities from above).

**As per claims 52-54**, refer to claims 13-15 for corresponding rejections; and Underwood: Fig. 93A-B; col. 252, lines 21-33.

**As per claim 55**, this claim include similar limitations as in claim 17, hence is rejected herein using the corresponding rejection as set forth therein.

**As per claim 56**, this is a medium version of claim 51, hence incorporates the same rejection as set forth in claim 51 above.

Art Unit: 2124

**As per claim 57**, Underwood discloses a method for developing software in a programmed digital computer, comprising: receiving a request for merging development documents (e.g. col. 90, line 59 to col. 91; Fig. 106-118 – Note: merge command is implicit to PVCS commands); selecting one of a plurality of merge preprocessors (see claim 29); modifying at least one development documents in the selected preprocessor (see claim 28); merging the development documents in a merge engine to produce a merged document (see claim 28).

**As per claim 58**, see claim 38 and 37 respectively.

**As per claim 59**, see claim 36.

**As per claim 60**, Underwood discloses documents with content and properties ( e.g. *Properties /scroll down 11400* - Fig. 114 – Note: content is inherent to any source file).

**As per claim 62**, this is the medium version of claim 57, hence incorporates the same rejection as set forth in claim 57 above.

### ***Claim Rejections - 35 USC § 103***

4. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

5. Claims 11, 39, and 40 are rejected under 35 U.S.C. 103(a) as being unpatentable over Underwood, USPN: 6,523,027, in view of (no author) Derwent 199741, JP- 09198393, “Document searching apparatus in database”, Pub. Date: July 31, 1997, IPC: G06F 017/30 ( hereinafter JP-DW-1997).

**As per claim 11**, in reference to claim 1, Underwood only discloses a SourceSafe tool with ways to administer documents or construct a header using a set of keywords (e.g. Fig. 109; col. 170, line 55 to col. 171, line 10; col. 183, lines 40-43) or COM interface using keyword (*ByRef* - col. 298, lines 24-45); but does not specify a keyword expansion unit. The use of keyword analysis in a search engine associated with COM and database engine such as suggested by Underwood was a known concept in the art of search engine. In a similar database query method analogous to database search and administrating of documents by Underwood, JP-DW-1997 discloses a keyword expansion unit ( Derwent excerpt: front page). It would have been obvious for one of ordinary skill in the art at the time the invention was made to add to Underwood's versioning system the keyword expansion unit or service as taught by JP-DW-1997 because this expansion would expand the search criteria input and improves search efficiency and adaptation rate.

**As per claim 39**, this claim incorporates the same limitations as addressed in claim 11. Hence, the limitations of receiving a expansion command and selecting of 'providers' in response thereof, a keyword broker and plurality of expanders would also been obvious using the rationale set forth in claims 28 and 29, because receiving a expansion command is implicitly disclosed in the combined teachings used therein.

**As per claim 40**, Underwood discloses keyword processing for COM and object querying and JP-DW-1997 discloses expansion of keyword for search. Underwood, however, specifies domain differential between objects and components assembled in the modeling scheme (e.g. 342, lines 20-29), hence suggests a domain associated with objects to retrieve from the objects reuse repository. It would have been obvious for one of ordinary skill in the art at

Art Unit: 2124

the time the invention was made to associate a domain (as suggested by Underwood) to the keyword expansion as taught by JP-DW-1997 and combine such expansion with Underwood's COM querying service because this domain defines better the object to be retrieved and enhances the expediency with which the keyword expansion is operable to help such query as taught by Underwood.

6. Claim 61 is rejected under 35 U.S.C. 103(a) as being unpatentable over Underwood, USPN: 6,523,027, as applied to claim 60, in view of (no author) Kramer, USPN: 6,216,140 (hereinafter Kramer).

As per claim 61, Underwood discloses merging of content (e.g. col. 90, line 59 to col. 91, line 14; *NoActionsMerge* - col. 258, lines 50-66) but does not specify merging separately from properties. Kramer, in a development documents merging method analogous to Underwood's, discloses merging with separation of attributes and other properties (e.g. *limited value attributes* - col. 11, line 15 to col. 12, line 31). It would have been obvious for one of ordinary skill in the art at the time the invention was made to add to Underwood's merging technique the merging of attributes as suggested by Kramer because, according to Kramer, "more inclusive or conservative of the changes between source and target versions is favored".

#### ***Response to Arguments***

7. Applicant's arguments filed 12/31/2003 have been fully considered but they are not persuasive. Following are the reasons therefor.

(A) As per claim 1, Applicants argued that Underwood does not teach or suggest that the 'activity' routes commands for execution on development documents from the version store (Appl. Rmrks, pg. 14, bottom, pg. 15, top 3 para). The rejection points to a server-client

Art Unit: 2124

paradigm wherein services like database services are sought from the developer workstations. Thus, the routing of commands from Web-based applications to develop or formulate business requirements to such database storing versioned objects are done via views and activity processes. The inherent nature of routing from where the request for retrieving persistent objects stored on network based servers are disclosed in the web protocol, with the commands from user or developer to be sent for data request being implicitly disclosed in window-based and web processing interface. The cited portions or drawings show plenty of evidences of such instances of requests or commands to be sent over remote services for objects retrieval or services.

(B) As per Applicants about same compatible interface (Appl. Rmrks, pg. 15, bottom, pg. 16, top 3 para), the rejection has now addressed how such limitation has been interpreted and addressed. It is not specified from the claim what a mutually compatible interface according to Applicants' view constitutes of; and Applicants fail to show how the Underwood's servers and client machines intercommunicating via an activity interface and using Web-based user interface as cited in the rejection, fail to be interfaces that cannot mutually communicate with each others given the explanation from the rejections.

(C) As per claim 2, Applicants argued that backup router and Altavista do not teach what is claimed as third party service providers (Appl. Rmrks, pg. 16, bottom, pg. 17, top 3 para). The rejection has explained that Underwood's Reta non-original products used for providing services to support Underwood's Reta framework are considered third party products used to replace what original services Reta cannot provide.

(D) As per claim 4, Applicants have submitted that the cited portions do not teach set of methods as API (Appl. Rmrks, pg. 17, 2<sup>nd</sup> para) but the rejection has shown set of APIs from

Art Unit: 2124

database services wherein methods to effect for objects query; and there is no explicit details in the claim that help otherwise distinguish what Applicants believe to be the claimed features from what has been cited in the rejection.

(E) As per claim 5 (Appl. Rmrks, pg. 17), Applicants' arguments are referred back to section A from above.

(F) As per claim 6, the claim recites 'where the API includes at least a subset of OLE DB' (Appl. Rmrks, pg. 17). There are no explicit details as to distinguish the claim from what has been cited in the rejection wherein Underwood teaches using databases API set of methods and OLE DB being part of the Oracle DB connectivity or ODBC.

(G) As per claim 7, Applicants revisit the issue about compatible interface (Appl. Rmrks, pg. 17, bottom; pg. 18, top) in the context of services providers. As has been mentioned in section B, the rejection again has shown examples of user interfaces from the services (e.g. Rationale Rose modeling service) to provide source control and modeling tool with connectivity to remote services like persistent storage of reuse objects or versioned source data. The arguments fail to put forth how the cited portions fail to read on what has been recited.

(H) As per claim 8, (Appl. Rmrks, pg. 18), the rejection has shown a command line utility and more examples of lines in which a user command can be generated. Applicants' remarks fail to show in which way what is recited in the claim distinguishes over the prior art, hence amount to mere allegations.

**As per claims 12-19 and 21-24:**

(I) As per claim 12 argument (Appl. Rmrks, pg. 18), please refer to section A.



Art Unit: 2124

(J) As per claims 15,16, and 18, (Appl. Rmrks, pg. 18, bottom ), the rejection has provided a query processing, an enlistment management, an interface to hold and retrieve documents in the light of a mutually compatible interface as previously addressed. Applicants' remarks fail to show in which way what is recited in the claim distinguishes over the prior art, hence amount to mere allegations.

(K) As per claim 17 (Appl. Rmrks, pg. 19, 2<sup>nd</sup> para), Applicants fail to provide from what is claimed how a development documents teaches away from the versioned source data stored and retrieved in Reta framework as cited from Underwood's reference.

(L) As per claims 21-24, Applicants' remarks fail to show in which way what is recited in the claims distinguishes over the prior art, hence amount to mere allegations.

**As per claims 25-31, 33:**

(M) The arguments about service providers and compatible interfaces (Appl. Rmrks, pg. 19, 2<sup>nd</sup> half )have been addressed in section A and B from above.

**As per claims 34-38:**

(N) Applicants have submitted that there is no explicit disclosure of merge engine (Appl. Rmrks, pg. 20, 2<sup>nd</sup> para). The rejection has pointed to a tool where development documents are versioned, updated, and integrated among different check-in check-out instances by developers, and a merge command as an option to reconcile checked out versioned documents. There is no details from the claim that can distinguish what is recited by the rejection from it is actually recited because a 'merge engine' is interpreted as a software capability operating on merging documents in situations as described above in Underwood's Version control tool. Applicants'

Art Unit: 2124

remarks fail to show in which way what is recited in the claims distinguishes over the prior art, hence amount to mere allegations.

(O) Applicants have submitted that there is no teaching or suggestion of ‘merge broker’ (Appl. Rmrks, pg. 20, 3<sup>rd</sup> para). The rejection has shown evidence of a merge engine; and of what represent a plurality of preprocessing elements from earlier claim rejections, e.g. claims 28-29, that discloses effects of brokers or preprocessors or subprocessors. The claim reciting the limitation ‘preprocessor’ or ‘broker’ do not explicitly describe what exactly defines and constitutes such limitation, therefore a broad and reasonable interpretation thereof has been used. Besides, Applicants do not provide sufficient support in traversing those claims, hence the arguments about a ‘merge broker’ amount to mere allegations without support as to how the combination of the prior art teachings as shown from the rejection would be inappropriate or teach away from what is disclosed.

(P) Applicants further argued that a merge engine does not exist (Appl. Rmrks, pg. 20, bottom para). This has been addressed in section N above.

(Q) As per claim 36, Applicants have submitted that there is no teaching of a group of preprocessors (Appl. Rmrks, pg. 21, top half). As earlier mentioned, the preprocessor limitation has been interpreted as processing element preceding another processing element, using the broadest interpretation. The rejection has pointed to a command line in a Version control Tool user interface, such teaching entailing a sequence of action submission to processing elements in light of processing sequence (*Workbench, SourceSafe Administrator, SourceSafe Explorer* – Fig. 104-117) before the document gets reconciled in the version store during its usage in Underwood’s Reta development process. The sequence of processing elements and selection of

Art Unit: 2124

which processing elements are implicitly disclosed in view of the commands generated from the user interface, according to the interpretation as set forth in the rejection of claim 29 above.

**As per claims 41-49 and 50:**

(R) Applicants arguments (Appl. Rmrks, pg. 21, bottom half) have been addressed correspondingly in the above sections. Besides, Applicants' remarks fail to show the manner in which what is recited in the claim distinguishes over the prior art, hence amount to mere allegations.

**As per claims 41-49 and 50 and claims 57-60, and 62:**

(S) The arguments presented (Appl. Rmrks, pg. 22, bottom half; pg. 23, top) here have been addressed in the corresponding sections from above. Again, Applicants' remarks fail to show in which way what is recited in the claim distinguishes over the prior art, hence amount to mere allegations.

**As per claims 11, 39, 40:**

(T) As for arguments on features on claims 11, 39, and 40, some of these relate to features that have been addressed earlier.

As for claim 39 (Appl. Rmrks, pg. 23; bottom half, pg. 24, 2<sup>nd</sup> para), the plurality of processing units and use of keyword expansion module are limitations that have been addressed in claim 11. The limitation as to 'select one of the providers' is subject to interpretation that 'providers' is not definite and can represent any service and there is no indication from the claim that such providers are necessarily keyword expansion providers or else. A potential USC 112 2<sup>nd</sup> paragraph rejection would be further considered only upon allowance of the claim. The reference used to provide a known concept such as to implement a keyword expansion unit in a database

Art Unit: 2124

search engine has been used via the Derwent 199741 reference. The limitations about selecting of providers to process a command have been addressed in claims 28-29. Applicants fail to point out how the combination as set forth in the rejection is not proper or would teach away from the respective teachings used. The rejection is founded on a combination based on Underwood's suggestion to provide a plurality of processing elements to process a database query search, on the well-known practice to implement search keyword and to adopt expansion units in such search operation by Derwent. Applicants do not show how such obvious combination is inappropriate; but keep asserting that none of the reference taken separately does teach the claimed features.

***Allowable Subject Matter***

8. Claims 63-66 are allowed.

The allowed feature is 'selecting one of the keyword expanders' (claim 63) or 'selecting one of a plurality of keyword-expansion service providers' (claim 66).

9. Claim 32 is objected to as being dependent upon a rejected base claim, but would be allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claims.

The allowable feature is 'a keyword broker for selecting among the expanders'.

***Conclusion***

10. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after

Art Unit: 2124

the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Tuan A Vu whose telephone number is (703)305-7207. The examiner can normally be reached on 8AM-4:30PM/Mon-Fri.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kakali Chaki can be reached on (703)305-9662.

**Any response to this action should be mailed to:**

Commissioner of Patents and Trademarks

Washington, D.C. 20231

**or faxed to:**

(703) 872-9306 ( for formal communications intended for entry)

**or:** (703) 746-8734 ( for informal or draft communications, please label

“PROPOSED” or “DRAFT” – please consult Examiner before use)

Hand-delivered responses should be brought to Crystal Park II, 2121 Crystal Drive, Arlington, VA. , 22202. 4<sup>th</sup> Floor( Receptionist).

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is (703) 305-3900.

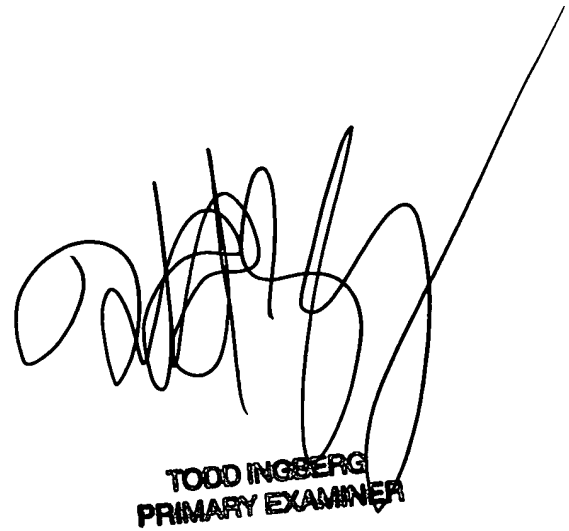
Application/Control Number: 09/717,537

Page 21

Art Unit: 2124

VAT

March 14, 2004



**TODD INBERG**  
**PRIMARY EXAMINER**